Network Planning for Disaster Recovery

Andrea Bianco*, Jorge Finochietto†, Luca Giraudo*, Marco Modesti*, Fabio Neri*
* Dip. di Elettronica, Politecnico di Torino, Italy, Email: {firstname.lastname}@polito.it
† Universidad Nacional de Cordoba - CONICET, Argentina, Email: jfinochietto@uncor.edu

***Abstract*—Disaster Recovery and Business Continuity issues are becoming fundamental in networks since the importance and social value of digital data is continuously increasing. On the one hand, there is an obvious need of backing up data for resilience against major failures; in many situations the process of storing backup data is also enforced by the law. On the other hand, providing services that allow the migration of applications in real-time through virtualization techniques is becoming a mandatory feature in several business situations.**

In this paper we analyze the problems and the challenges of off-site data replication and virtual machine migration. In particular, we discuss the issue of optimizing network planning to support disaster recovery and business continuity.

ILP (*Integer Linear Programming*) formulations for the optimization problem are presented with different objective functions. Heuristics are also proposed and analyzed taking into account both network cost minimization and fault recovery efficiency.

I. Introduction

After the events of September 11^{th} 2001, the interest in *Disaster Recovery* techniques has grown sharply. Disaster Recovery refers to all the activities and policies implemented in a wide range of applications to recover resources (e.g. stored data, communication links, switching points, business processes) after an outage event induced by natural or human factors. Among the activities comprised in the Disaster Recovery framework, *Off-site Data Protection* is the process of copying critical data to a physically remote site, where storage resources are available. Today, the most widely used solutions to backup data rely on the combination of two technologies: RAID [1] and Fibre Channel [2].

Fibre Channel is a network architecture designed for SANs (Storage Area Networks) to interconnect servers and storage devices in a fast and reliable manner. While originally designed for utilization inside data centers, it can be adopted over geographical distances, as demonstrated by commercial products currently offered by major vendors. It implements complex link-by-link flow control algorithms to control congestion and to avoid packet losses. The Fibre Channel technology is not based on the well-established TCP/IP/Ethernet suite; thus, it is relatively expensive because of lower production volumes and of the utilization in the physical layer of high-end transceiver and expensive mono-modal fibers. Fiber channels also appears to be more complex to manage if compared to traditional network technologies like IP and Ethernet. To extend FC capabilities over IP networks, solutions like Fibre Channel over IP (FCIP) and Internet Fibre Channel Protocol (iFCP) have been defined by the Internet Engineering Task Force (IETF).

Recently, IETF has standardized the iSCSI protocol [3], a SAN protocol based on TCP/IP that may become an alternative to Fibre Channel. SCSI (Small Computer System Interface) is a widely deployed interface between computers and directly attached storage devices. iSCSI transports SCSI commands issued to disks over TCP connections, to permit reading/writing from/to remote devices. Indeed, iSCSI enables off-site data protection over traditional LAN (Local Area Network), MAN (Metropolitan Area Network) and even WAN (Wide Area Network) technologies. Thus, iSCSI may be a convenient alternative to Fibre Channel, since it does not require the deployment of a separate infrastructure for SANs, nor the management of a different network.

An alternative to iSCSI is the DRDB (Distributed Replicated Block Device) protocol, an open-source project that provides transparent replication of hard-disk's content at the block level using the standard TCP/IP network stack. Since both iSCSI and DRDB are based on remote transport over TCP/IP, packet losses and retransmissions may occur and large latencies may be suffered. Thus, in general, they provide lower performance than the high-end SAN architectures in which SCSI commands are transported by Fibre Channel.

While combinations of the above technologies enable storage virtualization, application/server virtualization solutions like XEN or VMware are mature enough to be employed in mission critical systems. As an example, advanced mechanisms like *live migration* of virtual machines are already stable and usable. This feature is especially interesting for disaster recovery applications, since a virtual machine can be moved transparentlyamong sites, if a short idle period of few seconds is acceptable.

Considering the requirements of disaster recovery and the features of products for both server and storage virtualization, we discuss in this paper an optimization problem, taking into account both remote storage needs and virtualization techniques. We will mainly refer to the Linux/RAID/iSCSI/TCP/IP configuration, but most of the presented considerations and design techniques hold for the SCSI/Fibre Channel protocol configuration. Thus, each site hosts virtual machines and it makes disk resources available through iSCSI over an underlying network that connects all sites. Virtual machines provide logical services and they may migrate among sites in case of failures. To provide business continuity, each virtual machine running on a site is connected to a local disk and a backup one in a remote site. Data are assumed to be copied transparently using RAID controllers on the local and remote disks through iSCSI. Therefore, if an outage event occurs the

Proceedings of the 2008 16th IEEE Workshop on Local and Metropolitan Area Networks
978-1-4244-2028-5/08/$25.00 ©2008 IEEE

virtual machine can be restarted on the remote site.

The paper aims at discussing different approaches to optimize virtual machine placement and disk assignments. These approaches aim at mitigating the impact that the network would suffer from after an outage event due to the reassignment of virtual machines to provide business continuity. To this aim, ILP (*Integer Linear Programming*) formulations are presented to describe different optimization scenarios, which take into account network performance, migration costs, and backup service quality. Finally, heuristics are proposed to solve these optimization problems and their performance are assessed through simulation.

II. Problem Definition

The ultimate goal of our work is to distribute efficiently storage resources (*disks*) and computing resources (*virtual machines*, VMs) among network nodes (*sites*) considering different optimization goals. The underlying network topology is assumed to be known and links have infinite capacity. More formally, we denote by:

S	the set of sites,
V	the set of virtual machines,
D	the set of disks,
v	the index of the set of virtual machines,
d	the index of the set of disks,
$s(v)$	the site where the v-th virtual machine runs,
$x_{v,d}$	a binary variable, set to 1 if the v-th VM is associated with the d-th disk, and to 0 otherwise.

We define the following logical constraints for the assignment problem.

- **One local disk per VM:** Every VM must be associated with one disk residing at the same site where the VM is hosted.
- **One remote disk per VM:** Every VM must be associated with one (backup) disk residing at a different site from the one where the VM is hosted.
- **One VM per disk:** Every disk must be associated with at most one VM (i.e., disks can not be shared among VMs).

Thus, every VM must be associated with two disks to guarantee a backup copy for the application data. Note that disks do not refer to physical resources, but rather to a disk service quantum needed by each VM to properly run on both the local and the remote site. All VMs are not assumed to be alike, i.e. they may require a different access speed (also named bandwidth request) to disks. In the ILP formulation, to speed up the solution process, the constraints are formalized as reported in Table I.

We assume that VMs are running on pre-defined sites: VMs are randomly assigned to sites in a pre-allocation phase which is not taking into account any optimization issue. Also, each VM is associated with one local disk at the site hosting the VM. As such, the optimization problem considers as unavailable the local disk associated with VMs and takes into account only available disks.

Sites are distributed according to a known underlying LAN/MAN/WAN network topology. Optimal paths used by routing algorithms are assumed to be known, and are used either a-priori as input parameters in the optimization problem or a-posteriori to measure the assignment quality in terms of network performance if the network topology is not considered in the optimization problem.

TABLE I
ILP COMMON CONSTRAINTS

One local disk per VM	**One remote disk per VM**
$\sum_{d \in s(v)} x_{v,d} \geq 1, \qquad \forall v \in V$	$\sum_{d \notin s(v)} x_{v,d} \geq 1, \qquad \forall v \in V$
One VM per disk	**Two disks per VM**
$\sum_{v} x_{v,d} \leq 1, \qquad \forall d \in D$	$\sum_{d} x_{v,d} = 2, \qquad \forall v \in V$

III. Problem Formulation

In this section we report the ILP formalization of the optimization problem considering different types of objective functions. These objective functions are introduced in the ILP formulation to address a particular metric that we would like to optimize. Indeed, when trying to find an optimal disks to VMs assignment, several metrics can be taken into account to assess the optimality of the solution. As such, in the following subsections, we present four different optimization problems that give raise to different disks to VMs assignments.

A. First model: network wide bandwidth optimization

One of the main problems when allocating backup disks in remote sites is the performance limitation that could be faced by running VMs that access remote storage resources. Indeed, iSCSI may provide a very limited throughput in presence of large delays, as discussed in [4], [5].

To control network congestion and to limit (indirectly) the maximum delay between VMs and remote disks, an obvious approach is to minimize the maximum bandwidth utilization on the underlying network links.

To formalize this optimization problem, we introduce a set of new variables, denoted by y_{vdhk}, where v is the VMs index, d is the disk index, and h, k represent sites s_h and s_k. More precisely, y_{vdhk} is a binary parameter set to 1 if and only if the link connecting sites s_h and s_k is used by the data flow from/to VM v and backup disk d, according to the decision made by the routing algorithm. Furthermore, we denote by B_v the bandwidth demand of the VM v when accessing its remote backup disk.

The objective function of this model is expressed as follows:

$$\min \max_{h,k} \sum_{v,d} B_v \times y_{vdhk} \times x_{vd}$$

B. Second model: network wide hop count optimization

An alternative approach to bound the end-to-end delay of each connection, thus limiting the penalties introduced by the iSCSI protocol in the presence of large delays, is to minimize the maximum number of hops in the path between the VMs and their backup disks. Thus, the goal is to assign a remote disk d to each VM v such that the sum of the hop counts is minimized.

Let h_{vd} be the number of hops between VM v and disk d. Since the assigned disk must be located in a different site, $h_{vd} = \infty$ when VM v and disk d reside on the same site.

The objective function is as follows:

$$\min \sum_{v,d} h_{vd} \times x_{vd}$$

C. Third model: service wide optimization

The two previously proposed models consider mainly networking performance, and assume that the system is stable and that VMs run at fixed positions assigned prior of running the optimization problem. Let us assume that one of the sites crashes. In several practical situations, the crash will not be instantaneous, and time may be available to migrate the applications active at the crashing site. Alternative situations can be envisioned, e.g., copying periodically the applications image to the backup site, in which a copy of the application can be restored after the crash, starting from the most recent state. In all these case, it would be important to minimize the overload caused by such a disastrous event. More precisely, when a site crashes, all the VMs running on it must migrate and/or should be restarted at a different site to keep their normal operation. A reasonable choice would be to migrate the VMs to the site hosting their backup disk to optimize performance. This migration process is a CPU-consuming activity on the remote site; therefore, it might slow down VMs already running on the backup sites.

To limit the maximum amount of CPU consumption needed to complete the migration process, i.e. to minimize the number of VMs that need to be restored in a given site, a new objective function is introduced:

$$\min \max_{h,k \in S} N_{hk}$$
$$N_{hk} = \sum_{v \text{ in } s_h, d \text{ in } s_k} x_{vd}$$

where the variable N_{hk} represents the number of new VMs that site s_k must initialize when site s_h crashes, and the notation v in s_h represents VM v hosted in site s_h. The variable N_{hk} represents also the number of disks residing at site s_k belonging to VMs hosted by site s_h.

D. Fourth model: constrained service wide optimization

This model combines the features of the previous two models. First, a maximum hop count is defined to connect VMs and disks to limit performance degradation due to delays introduced at each hop. Then, this value is used as an additional constraint to run the service-wide optimization problem (third model). In other words, we restrict the distribution of backup disks to the "nearby" sites, that is, sites whose distance is bounded by the maximum number of hops.

The new constraint is expressed as follows:

$$x_{vd} \times h_{vd} \leq max\_hop \qquad \forall v, d$$

Note that when VM v and disk d are not associated, this constraint always holds, since $x_{vd} = 0$ and $max\_hop \geq 0$.

IV. Heuristics

These proposed ILP models can be demonstrated to be NP-Hard since they are actually extensions of the well-known assignment problem [6]. Therefore, we need to evaluate heuristics that can run in polynomial time. In this section we define some heuristics that can be used to solve these problems.

A. Longest Processing Time (LPT)

The (LPT) Longest Processing Time problem [7] is a scheduling algorithm devised in the field of operating systems design: the idea is to sort CPU's jobs in decreasing order of processing times, serving the largest job first.

This idea can be simply adapted to our scenario: serve VMs in order of largest bandwidth request, assigning to VMs the disk reachable via the least occupied path. This balance the load on the path, providing on average the best disk access performance to VMs.

The main steps of the LPT algorithm are:

1) Disk pre-allocation: every VM is associated with a local disk in the site hosting the VM. The disk is removed from the list of available disks.
2) While there is a VM without an assigned backup disk:
    a) Select the VM with the maximum bandwidth request.
    b) Search the available disk which can be reached through the path with the maximum available bandwidth.
    c) If a disk is found, assign the disk to the VM, and go to the next VM. Otherwise, signal that the procedure was unable to find a complete assignment.

The LPT approach described in this section is a *greedy* algorithm. Thus, first, there is no guarantee that the solution is a global optimum. Second, and most important, the algorithm may not converge to a complete assignment, i.e. a solution that finds a remote backup disk for all VMs, even if a complete assignment exists. The key factor is the distribution of available disks and virtual machines among sites. This problem is critical especially when the number of available disks is not large. Indeed, it could be impossible to find an available remote disk: this happens especially when the disks are not uniformly distributed, but are mainly concentrated in certain sites.

B. Minimum Weight Assignment (MWA)

The previously presented method takes into account the intrinsic limitation of storage protocols like iSCSI, that suffer performance degradation in the face of large delays, by heuristically minimizing the maximum load on network links, similarly to the first ILP. Another approach would be, as in the second ILP model, to find a solution that limits the hop count between a VM and its backup disk. As a side effect, this approach reduces the link bandwidth utilization because flow lengths are limited to few hops on average.

Let us consider a bipartite graph, where two sets of nodes exist, and edges can only connect nodes belonging to two different sets. All VMs are in the left-hand side set and all disks are in the right-hand side set. Then, to introduce the one-local-disk constraint, we remove from the right-hand side set all nodes (disks) assigned locally to VMs by the pre-allocation algorithm. The remaining edges between VMs and disks are assigned a weight equal to the number of hops in the path interconnecting the site hosting the VM with the site hosting the remote disk.

We wish to select edges connecting VMs and disks so as to minimize the summation of selected edge weights, i.e. a set of edges that minimizes the assignment cost in terms of number of hops. We are subject to the "matching" constraint: select at most one edge per VM and at most one edge per disk. The Hungarian algorithm [6], [8], [9] can be used to find this assignment with minimum weight.

C. Disaster recovery (DR)

In this section another approach to the assignment of backup disk to VMs is presented. The previous methods focus mainly on optimizing performance when considering the normal operation phase, when failures do not occur. With this heuristic, we address the case of disaster recovery, as in the corresponding ILP model in III-C.

When a site crashes, we wish to minimize the number of VMs that have to be restarted on another site to maintain active the offered services. Indeed, to quickly reactivate the service, and to avoid excessive slowdown of running services, it is important to reduce the number of VMs that must be restarted at remote sites. Virtual machines are restarted on the site where its backup disks is hosted, thus, "migrating" virtual machines to these sites. This does not introduce any additional VM-to-disk traffic in the network, unless a new remote backup site is defined, an issue not considered in this paper. The heuristic we adopt to pursue this goal is the following:

1) Disk pre-allocation: every VM is associated with a site and selects a local disk. The disk is removed from the list of available disks.
2) Random selection of a non-matched VM: suppose this VM is hosted in site s_a.
3) Criteria to select the disk for the chosen VM:
    - the disk is in site s_b, different from site s_a (remote disk constraint);
    - among all sites, choose the "optimal" site s_b as the one hosting the minimum number of backup disks associated with VMs running in site s_a;
    - if there are several equivalent "optimal" sites, select the site s_b that has the maximum number of available disks.
    - if no site is found, the procedure ends and signals that was not possible to find a complete assignment.
4) If not all VMs are matched, return to step 2), otherwise STOP.

Thanks to the constraints used in the selection of the "optimal" disk, backup disks for VMs hosted at a given site s_a are distributed among all available sites, reducing the additional load at remote sites in the case of a disaster event at site s_a. Note that this heuristic could be modified to target the ILP model in III-D, i.e., to consider as "optimal" sites only those whose hop count is equal or less to the max_hop parameter. However, we do not further pursue this approach in the paper.

D. Maximum Size Assignment (MSA)

All the above described heuristics may not be able to find a complete assignment, i.e., assigning a backup disk to all VMs. A relatively simple approach to obtain a disaster-tolerant assignment, i.e., a complete matching, is to run a Maximum Size Matching (MSM) algorithm on the previously defined bipartite graph, by setting all edge weights to 1. Indeed, the Maximum Size Matching is a well-known algorithm in graph theory, used to maximize the number of selected edges in bipartite graphs. This assignment problem can be solved by, e.g. the Ford-Fulkerson method [10], [11]. This approach is a simplification of the previously defined Minimum Weight Assignment and guarantees that a disaster tolerant solution is always found, if any is available.

E. Heuristic goals

In summary, the objective of the first proposed heuristic (LPT) is to minimize the network load choosing the least occupied path. A similar objective is pursued by the Minimum Weight Assignment, which instead minimizes on average the number of hops in the VM to disk path. These heuristics could be useful in a scenario where the owner of VMs and disks is not the owner of the network's, thus, to reduce bandwidth rental costs from a network provider. Instead, the objective of the Maximum Size Assignment heuristic is only to maximize the number of active VMs with a backup disk; this method does not consider at all the network infrastructure. This heuristic may be of interest when the same entity owns the network infrastructure and the virtualization infrastructure and the main goal is to satisfy the largest number of VMs. Finally, the disaster recovery heuristic tries to optimize the recovery speed in the presence of a failure event.

V. Performance results

In this section, we report the results of our simulations: ILP problems were solved utilizing the CPLEX library, meanwhile heuristics were evaluated through custom C programs.

A. Simulation parameters

In our simulations, we consider the following performance indexes:

1) **Maximum bandwidth (MB):** the bandwidth occupation registered on the most loaded link.
2) **Mean bandwidth (mB):** the mean bandwidth occupation among active links.
3) **Mean cost (mC):** the mean length (in number of hops) of the paths between VMs and backup disks.

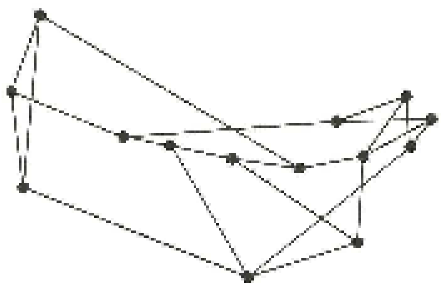

Fig. 1. The NSFNET topology considered in our simulations

4) **Maximum number of VMs (MV):** the maximum number of new virtual machines that a site must host when another site crashes.

The first three indexes are closely related to network performance: MB and mB are useful to estimate network congestion, while the mC index is an indication of the maximum end-to-end delay between VMs and disks. The last index mainly describes the optimality of the algorithm from the disaster recovery point of view. Note that, by minimizing the number of VMs that should be restarted at a given site upon failure, we also distribute the amount of traffic between the crashed site and all the involved backup sites during the reconfiguration phase.

The network topology used in simulations is derived from the *National Science Foundation Network (NSFNET)*, a well-known USA backbone network. This network comprises 14 sites connected by 22 links (Fig. 1). The shortest path algorithm is run as the routing algorithm to determine site-to-site optimal paths. However, under the hypothesis that VM migration should be as much transparent as possible, we assume that the migration is supported at layer 2 (e.g. Ethernet layer), to avoid IP address redefinition. In this case, all sites belong to the same extended LAN and to the same IP sub-net: as a consequence, the paths between sites are determined by the Prim's algorithm used to calculate the spanning tree. Thus, we also tested the proposed algorithms over a LAN/MAN meshed topology over which the spanning tree is running. Finally, we run experiments on a LAN/MAN ring topology. Even if the results have different absolute values, the general trends are very similar in all the above mentioned scenarios. As such, we report results for the NSFNET topology only.

All link capacities are assumed to be infinite. We evaluate the impact of the proposed approaches by examining the link load created in the underlying network. To emulate the behavior of different types of servers with different disk access rates, VMs are assigned a randomly chosen bandwidth request uniformly distributed between 10 and 100 Mbit/s. All disks and VMs are uniformly distributed among the 14 sites in the NSFNET topology. A total of 140 virtual machines (on average 10 VMs per node) are running on the network. The total number of disks available in the network ranges from a minimum value of 280 disks, corresponding to twice the number of virtual machines, to a maximum value of 560 disks.

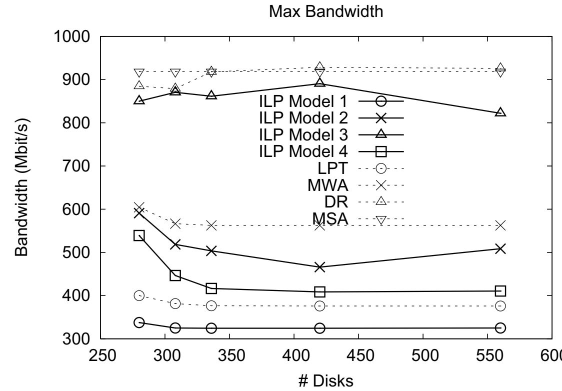

Fig. 2. Maximum link bandwidth

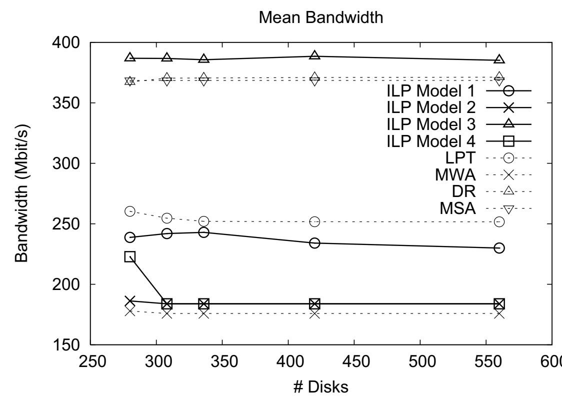

Fig. 3. Mean link bandwidth

Note that 280 disks is the minimum value needed to satisfy both the local disk and the backup disk constraints. Results are averaged over 20 network instances obtained by randomly assigning VMs and disks to sites.

B. Results

In Figs. 2 and 3, ILP model 3 and the two heuristics DR and MSA show the worst performance, since they do not take

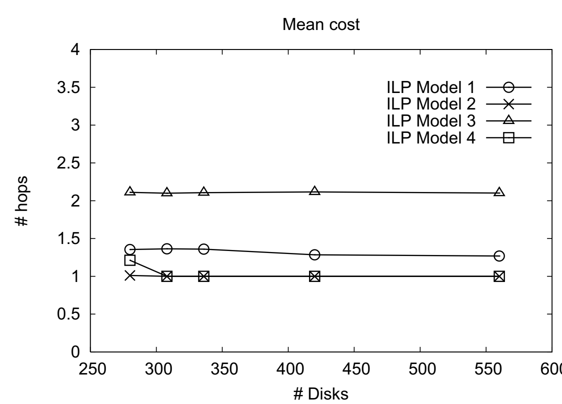

Fig. 4. Mean cost, i.e. the maximum path length in number of hops, for ILP models

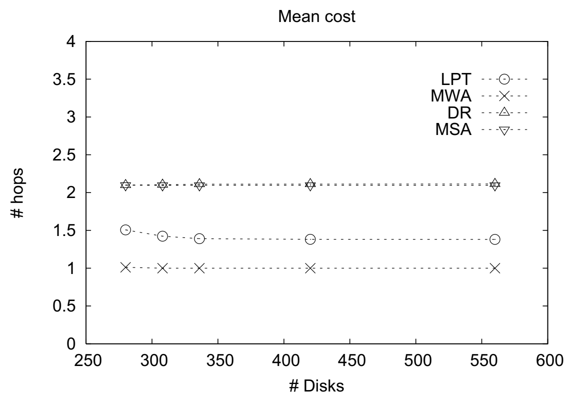

Fig. 5. Mean cost for heuristics

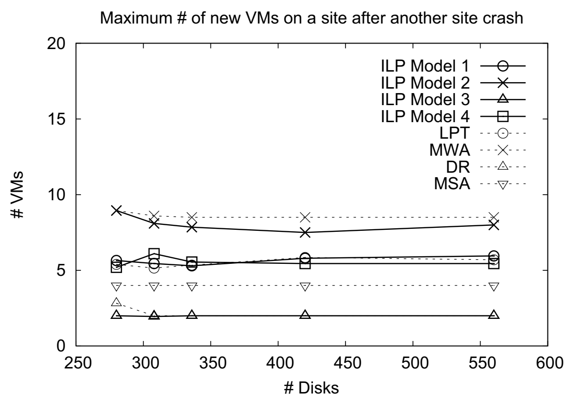

Fig. 6. Max number of new VMs that a node must host after node crash

into account network resources in their assignment. Indeed, they encourage the distribution of backup disks among all the available sites, thus selecting also paths with several hops. As shown in Fig.4 and 5, they exploit longer path in the network (the longest used path has 3 hops). ILP model 1 and the ”corresponding” LPT heuristics show the best performance in terms of maximum bandwidth, at the price of rather high mean bandwidth requirements. The best performance in terms of mean bandwidth utilization are provided by the ILP model 2 and the MWA heuristics, which however tend to create a higher load on the most loaded link. Both algorithms minimize the hop count. Thus, the minimum distance criteria is once again shown to be a sound network design approach. The ILP model 4 presents the best compromise when jointly considering both bandwidth performance indices.

Fig. 6 reports the maximum number of new VMs that must be restarted at a site when another site stops working. Obviously, algorithms trying to optimize network-wide performance require a higher maximum number of VMs, since they concentrate on a limited set of sites most of the backup resources needed by VMs hosted at the same site.

The ILP model 4 is a combination of a network-aware model and a disaster-recovery aware model. Thus, it obtains intermediate performance with respect to other models. From the network's point of view, it behaves similarly to the best models in this category, reducing the utilization of network resources. Furthermore, from the recovery's point of view, it behaves better than ILP models 2 and 3, although the required number of VMs is still larger than those required by the disaster-recovery oriented model.

All the heuristics show performance indices only slightly worse than those obtained when running the ”corresponding” ILP model. LPT and MWA show the best results from the network point of view. Meanwhile, the DR heuristic is a good approximation for ILP model 3 but it requires many network resources. LPT shows the best compromise between network and disaster recovery performance: a relatively low network occupation and a reasonable low number of VMs that should be restarted in the worst case after a disaster.

VI. Conclusion

We described several ILP models and heuristics algorithms to optimize VMs to backup disks assignment in WAN/MAN/LAN networks. We showed that well-known algorithms can be adapted to devise heuristic solutions to Disaster Recovery problems.

We evaluated the impact of both ILPs and heuristics on the underlying network and on the speed of the VM recovery process. Network performance and recovery speed are obviously contrasting goals: thus, a trade-off between network-wide and disaster recovery oriented performance indices is needed. The ILP model 4 and the LPT heuristics seems to provide the best compromise between these two contrasting goals.

Acknowledgments

These activities were developed in the framework of the FEDERICA project, funded by the European Commission.

References

[1] D. A. Patterson, G. A. Gibson, R. H. Katz, *A Case for Redundant Arrays of Inexpensive Disks (RAID)*, Proceedings of the International Conference on Management of Data (SIGMOD), June 1988.

[2] ANSI INCITS 373-2003, *Fibre Channel - Framing and Signaling (FC-FS)*, rev. 1.90, April 2003.

[3] IETF, (2004, April), *RFC 3720 (iSCSI)*, [Online], Available: http://www.ietf.org/rfc/rfc3720.txt

[4] C.M. Gauger, M. Köhn, S. Gunreben, D. Sass, S. Gil Perez, *Modeling and Performance Evaluation of iSCSI Storage Area Networks over TCP/IP-based MAN and WAN networks*, Proceedings Broadnets 2005, Oct. 2005.

[5] A. Bianco, M. Modesti, F. Neri, J. M. Finochietto *Distributed Storage on Networks of Linux PCs Using the iSCSI Protocol*, Proceedings HPSR 2008, May. 2008.

[6] H. W. Kuhn, "The Hungarian Method for the assignment problem", *Naval Research Logistics Quarterly*, 2:83-97, 1955.

[7] B. Chen, A. P. A. Vestjens, *Scheduling on identical machines: How good is LPT in an on-line setting?*, Operations Research Letters, Volume 21, Issue 4, November 1997, Pages 165-169.

[8] H. W. Kuhn, "Variants of the Hungarian method for assignment problem", *Naval Research Logistics Quarterly*, 3: 253-258, 1956.

[9] J. Munkres, "Algorithms for the Assignment and Transportation Problems", *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32-38, 1957.

[10] L. R. Ford Jr., D. R. Fulkerson, "Maximal flow through a network', *Canadian Journal of Mathematics*, 399-404, 1956.

[11] L. R. Ford Jr., D. R. Fulkerson, "A simple algorithm for finding maximal network flows and an application to the Hitchcock problem", *Canadian Journal of Mathematics*, 210-218, 1957.